Massively Parallel Proof-Number Search for Impartial Games and Beyond

Tomáš Čížek, Martin Balko, and Martin Schmid

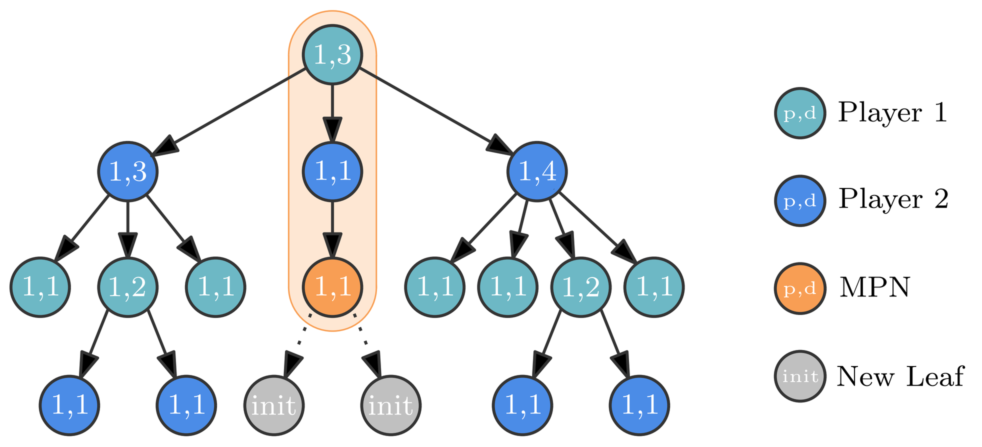
Department of Applied Mathematics, Charles University, Prague, Czech Republic

New parallel Proof-Number Search outperforms previous Sprouts solvers by 10,000×

Proof-Number Search (Allis, 1994)

Search algorithm for solving combinatorial games (Checkers). Expands *MPNs* towards **shortest proof** via proof numbers *PN*.

1) *Select MPN.*
2) *Expand MPN.*
3) *Estimate new PN.*
4) *Backpropagate PN.*

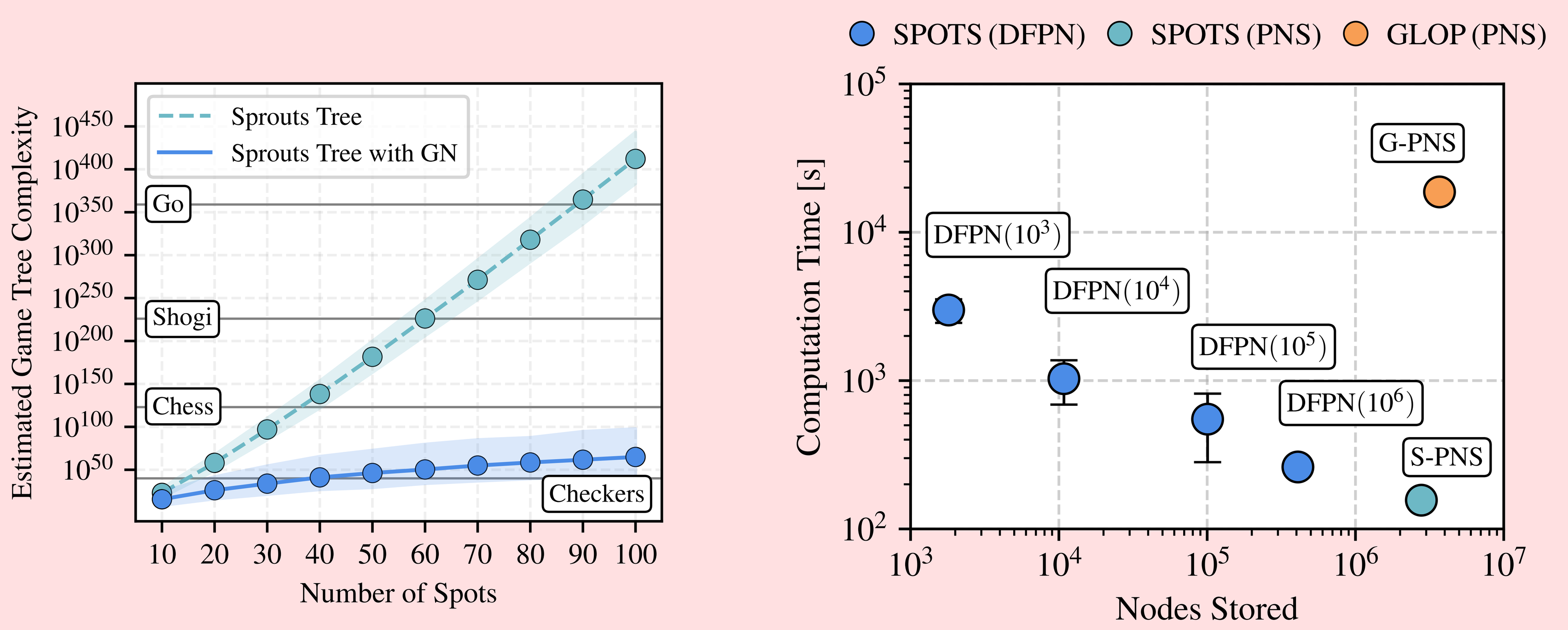


Impartial Games & Grundy Numbers

Combinatorial games with *identical moves* for both players. Grundy numbers allow **independent analysis** of subpositions.

DFPN Search with Grundy Numbers

Grundy numbers significantly **reduce search complexity** by extending classical AND/OR trees with *Grundy nodes*.

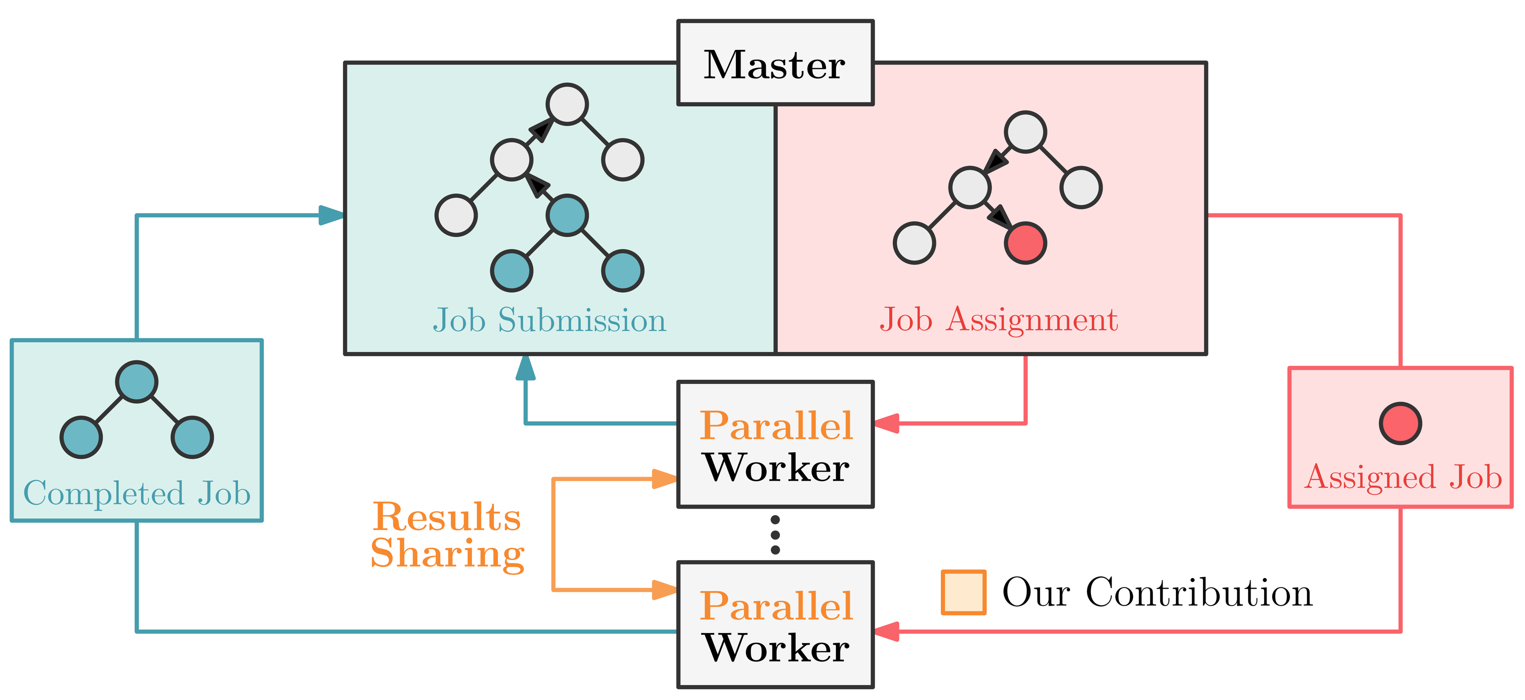


We **adapt DFPN**, a popular and memory-efficient variant of Proof-Number Search, to work with such game trees.

Seeking Parallel Proof-Number Search

Several studies explored parallel Proof-Number Search in diverse domains, but current approaches **scale poorly** and **saturate quickly**, achieving **speedups below 20×** on 64 cores.

Open problem: Kishimoto et al. (2012) highlighted the need for a well-scaling, massively parallel Proof-Number Search.

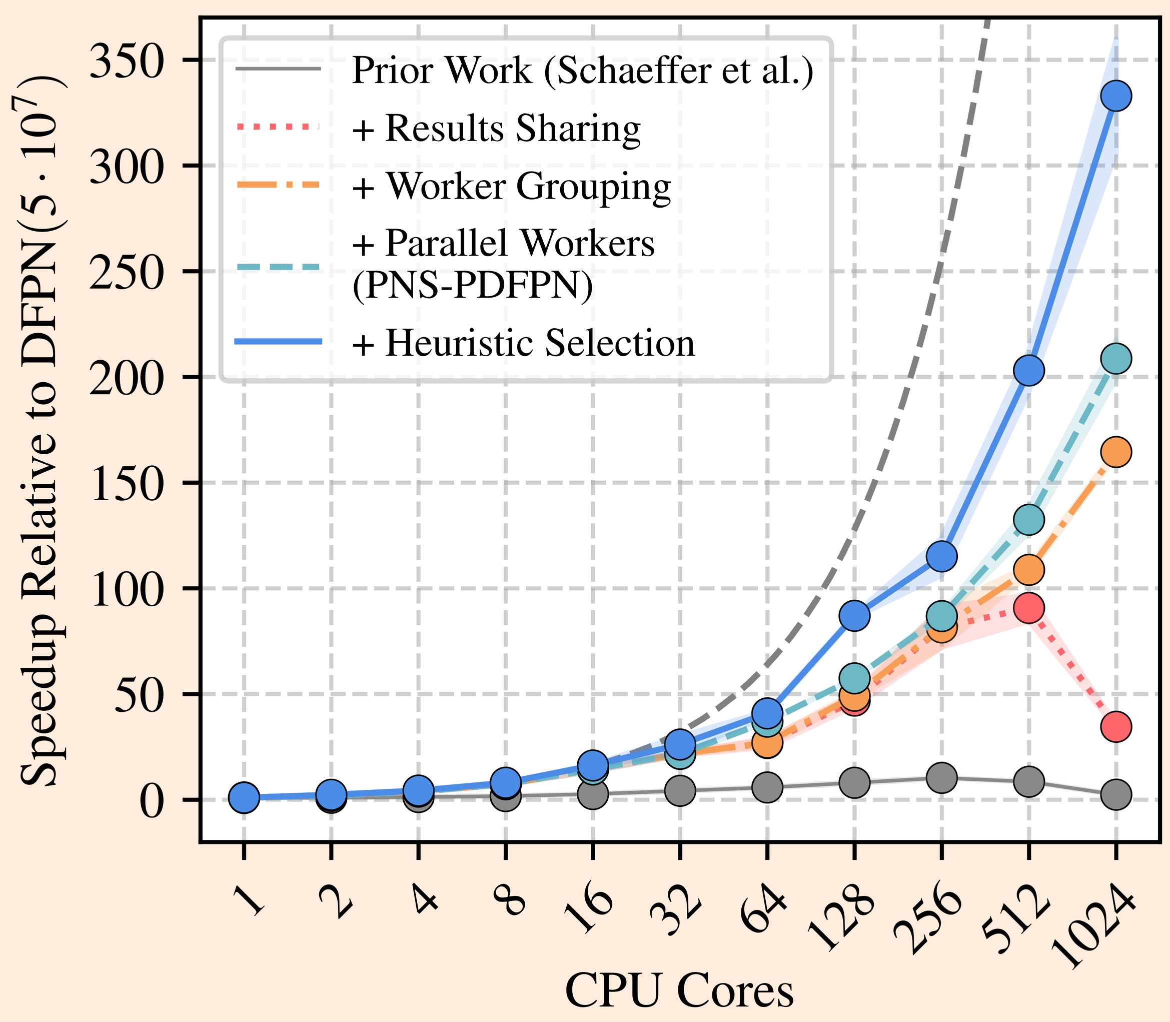


New Parallel Proof-Number Search

We present **PNS-PDFPN**, the first well-scaling parallel variant of Proof-Number Search for distributed-memory systems.

Key improvements:

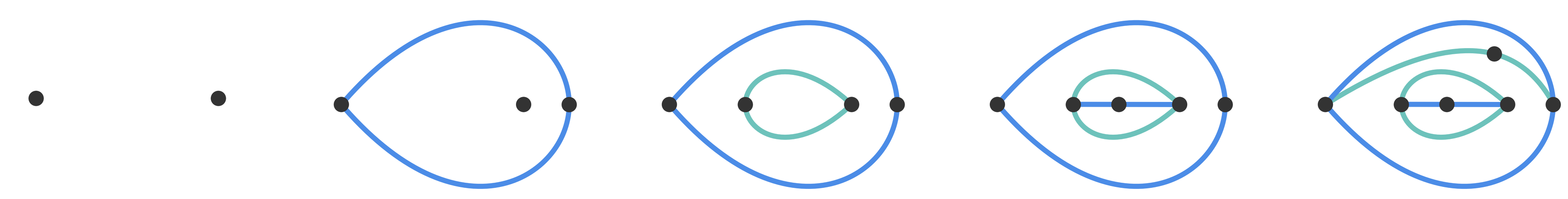
- **Sharing** computed outcomes between workers.
- Node-local worker **groups** with increased result sharing.
- **Second-level parallelization** of workers (parallel DFPN).
- Simple domain-specific **heuristic** for job selection.



Our algorithm achieves **speedup up to 332.9×** on 1024 cores.

Sprouts Game (Conway & Paterson, 1967)

Rules: Two players connect n spots with no-crossing curves; new spot along the curve; ≤ 3 incident curves; no move loses.
Complexity: Game tree complexity larger than for Chess or Go.



New Sprouts Results

Our parallel solver **SPOTS** computes outcomes of 42 new Sprouts positions with 10,000× speedup over prior work.

n	outcome	n	outcome	n	outcome	n	outcome	n	outcome
1	L	22	W	43	L	64	W	85	L
2	L	23	W	44	L	65	W	86	L
3	W	24	L	45	W	66	—	87	—
4	W	25	L	46	W	67	L	88	W
5	W	26	L	47	W	68	L	89	W
6	L	27	W	48	L	69	—	90	—
7	L	28	W	49	L	70	W	91	L
8	L	29	W	50	L	71	W	92	L
9	W	30	L	51	W	72	—	93	—
10	W	31	L	52	W	73	L	94	W
11	W	32	L	53	W	74	L	95	W
12	L	33	W	54	L	75	—	96	—
13	L	34	W	55	L	76	W	97	L
14	L	35	W	56	L	77	W	98	L
15	W	36	L	57	W	78	—	99	—
16	W	37	L	58	W	79	L	100	W
17	W	38	L	59	W	80	L	101	—
18	L	39	W	60	L	81	—	102	—
19	L	40	W	61	L	82	W	103	L
20	L	41	W	62	L	83	W	104	L
21	W	42	L	63	—	84	—	105	—

Conway · Mollison · Applegate et al. · Purinton · Lemoine and Vienot · Our Results

Supported by the grant no. 25-18031S of the Czech Science Foundation (GAČR), by the Charles University Grant Agency (GAUK) project no. 326525, and by the Charles University project UNCE 24/SCI/008. Computational resources were provided by EquiLibre Technologies, Inc. and by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

PAPER

GITHUB